Package: bnnSurvival (via r-universe)

August 26, 2024

Type Package

Title Bagged k-Nearest Neighbors Survival Prediction

Version 0.1.5

Date 2017-05-10

Author Marvin N. Wright

Maintainer Marvin N. Wright <marv@wrig.de>

Description Implements a bootstrap aggregated (bagged) version of the k-nearest neighbors survival probability prediction method (Lowsky et al. 2013). In addition to the bootstrapping of training samples, the features can be subsampled in each baselearner to break the correlation between them. The Rcpp package is used to speed up the computation.

Imports prodlim, pec, Rcpp (>= 0.11.2), parallel, methods

LinkingTo Rcpp

Suggests survival, testthat

License GPL-3

RoxygenNote 5.0.1

Repository https://mnwright.r-universe.dev

RemoteUrl https://github.com/mnwright/bnnsurvival

RemoteRef HEAD

RemoteSha 694072bc55d210dde9e3cccf03e1fe7a29dfd0e3

Contents

bnnSurvival *Bagged k-nearest neighbors survival prediction*

Description

Bootstrap aggregated (bagged) version of the k-nearest neighbors survival probability prediction method (Lowsky et al. 2013). In addition to the bootstrapping of training samples, the features can be subsampled in each base learner.

Usage

```
bnnSurvival(formula, data, k = max(1, nrow(data)/10),
  num_base_learners = 50, num_features_per_base_learner = NULL,
  metric = "mahalanobis", weighting_function = function(x) {     x * 0 + 1
  }, replace = TRUE, sample_fraction = NULL)
```

Arguments

`formula` Object of class formula or character describing the model to fit.

`data` Training data of class data.frame.

`k` Number nearest neighbors to use. If a vector is given, the optimal k of these values is found using 5-fold cross validation.

`num_base_learners`
Number of base learners to use for bootstrapping.

`num_features_per_base_learner`
Number of features randomly selected in each base learner. Default: all.

`metric` Metric d(x,y) used to measure the distance between observations. Currently only "mahalanobis".

`weighting_function`
Weighting function w(d(,x,y)) used to weight the observations based on their distance.

`replace` Sample with or without replacement.

`sample_fraction`
Fraction of observations to sample in [0,1]. Default is 1 for `replace=TRUE`, and 0.6321 for `replace=FALSE`.

Details

For a description of the k-nearest neighbors survival probability prediction method see (Lowsky et al. 2013). Please note, that parallel processing, as currently implemented, does not work on Microsoft Windows platforms.

The weighting function needs to be defined for all distances >= 0. The default function is constant 1, a possible alternative is w(x) = 1/(1+x).

To use the non-bagged version as in Lowsky et al. 2013, use `num_base_learners=1`, `replace=FALSE` and `sample_fraction=1`.

Value

bnnSurvivalEnsemble object. Use predict() with a new data set to predict survival probabilites.

Author(s)

Marvin N. Wright

References

Lowsky, D.J. et al. (2013). A K-nearest neighbors survival probability prediction method. Stat Med, 32(12), 2062-2069.

See Also

`predict`

Examples

```
require(bnnSurvival)

## Use only 1 core
options(mc.cores = 1)

## Load a dataset and split in training and test data
require(survival)
n <- nrow(veteran)
idx <- sample(n, 2/3*n)
train_data <- veteran[idx, ]
test_data <- veteran[-idx, ]

## Create model with training data and predict for test data
model <- bnnSurvival(Surv(time, status) ~ trt + karno + diagtime + age + prior, train_data,
                     k = 20, num_base_learners = 10, num_features_per_base_learner = 3)
result <- predict(model, test_data)

## Plot survival curve for the first observations
plot(timepoints(result), predictions(result)[1, ])
```

`get_best_k` *Get optimal number of neighbors*

Description

Get optimal number of neighbors for bnnSurvival by cross validation

Usage

```
get_best_k(formula, data, k, ...)
```

Arguments

`formula`	Formula
`data`	Data
`k`	Number of neighbors
`...`	Further arguments passed to bnnSurvival

Value

Optimal k

`predict,bnnSurvivalBaseLearner-method` *Compute prediction for all samples.*

Description

Compute prediction for all samples.

Usage

```
## S4 method for signature 'bnnSurvivalBaseLearner'
predict(object, train_data, test_data,
  timepoints, metric, weighting_function, k)
```

Arguments

`object`	bnnSurvivalBaseLearner object
`train_data`	Training data (with response)
`test_data`	Test data (without response)
`timepoints`	Timepoint to predict at
`metric`	Metric used
`weighting_function`	Weighting function used
`k`	Number of nearest neighbors

predict,bnnSurvivalEnsemble-method

Predict survival probabilities with bagged k-nearest neighbors survival prediction.

Description

Predict survival probabilities with bagged k-nearest neighbors survival prediction.

Usage

```
## S4 method for signature 'bnnSurvivalEnsemble'
predict(object, test_data)
```

Arguments

`object`	Object of class bnnSurvivalEnsemble, created with bnnSurvival().
`test_data`	Data set containing data to predict survival.

predictions

Get Predictions

Description

Get Predictions

Usage

```
predictions(object, ...)
```

Arguments

`object`	Object to extract predictions from
`...`	further arguments passed to or from other methods.

predictions,bnnSurvivalResult-method

Get Predictions

Description

Get Predictions

Usage

```
## S4 method for signature 'bnnSurvivalResult'
predictions(object)
```

Arguments

`object`	bnnSurvivalResult object to extract predictions from

predictSurvProb.bnnSurvivalEnsemble

Function to extract survival probability predictions from bnnSurvivalEnsemble. Use with `pec` package.

Description

Function to extract survival probability predictions from bnnSurvivalEnsemble. Use with `pec` package.

Usage

```
## S3 method for class 'bnnSurvivalEnsemble'
predictSurvProb(object, newdata, times, ...)
```

Arguments

`object`	bnnSurvivalEnsemble object.
`newdata`	Data used for prediction.
`times`	Not used.
`...`	Not used.

Value

survival probability predictions

`print,bnnSurvivalEnsemble-method` *Generic print method for bnnSurvivalEnsemble*

Description

Generic print method for bnnSurvivalEnsemble

Usage

```
## S4 method for signature 'bnnSurvivalEnsemble'
print(x)
```

Arguments

`x`	bnnSurvivalEnsemble object to print

`print,bnnSurvivalResult-method` *Generic print method for bnnSurvivalResult*

Description

Generic print method for bnnSurvivalResult

Usage

```
## S4 method for signature 'bnnSurvivalResult'
print(x)
```

Arguments

`x`	bnnSurvivalResult object to print

`show,bnnSurvivalEnsemble-method` *Generic show method for bnnSurvivalEnsemble*

Description

Generic show method for bnnSurvivalEnsemble

Usage

```
## S4 method for signature 'bnnSurvivalEnsemble'
show(object)
```

Arguments

`object`	bnnSurvivalEnsemble object to show

`show,bnnSurvivalResult-method` *Generic show method for bnnSurvivalResult*

Description

Generic show method for bnnSurvivalResult

Usage

```
## S4 method for signature 'bnnSurvivalResult'
show(object)
```

Arguments

`object`	bnnSurvivalResult object to show

`timepoints` *Get Timepoints*

Description

Get Timepoints

Usage

```
timepoints(object, ...)
```

Arguments

`object`	Object to extract timepoints from
`...`	further arguments passed to or from other methods.

`timepoints,bnnSurvivalResult-method` *Get timepoints*

Description

Get timepoints

Usage

```
## S4 method for signature 'bnnSurvivalResult'
timepoints(object)
```

Arguments

`object`	bnnSurvivalResult object to extract timepoints from

Index